

Fig 1

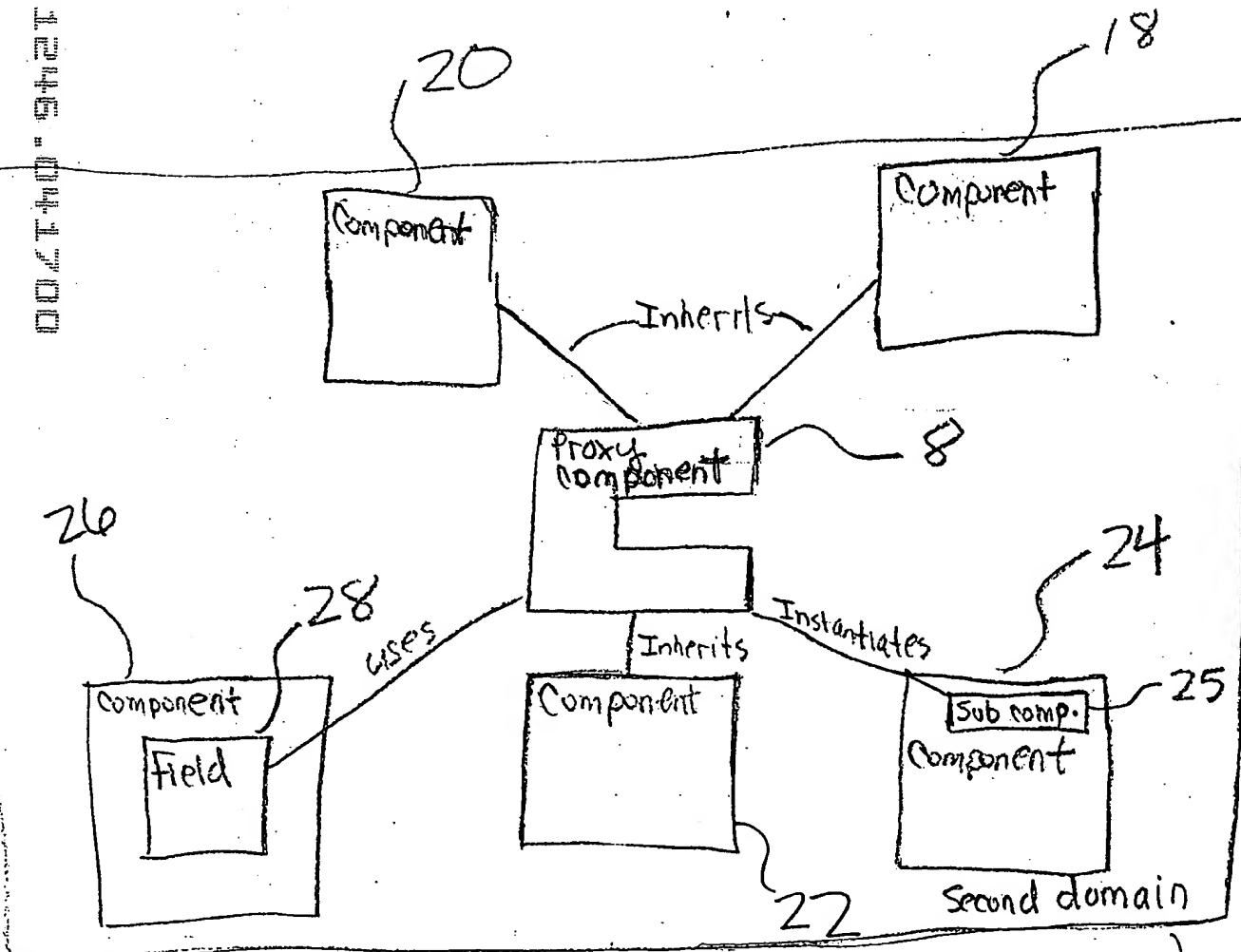


Fig 2

00551246-041700

Java

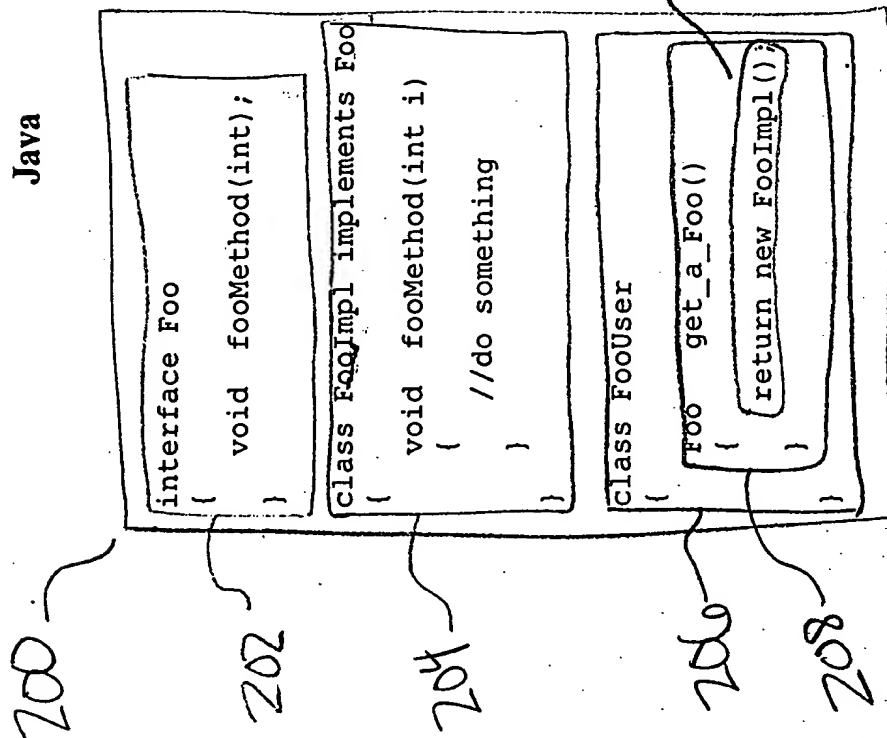


Fig. 3

C++

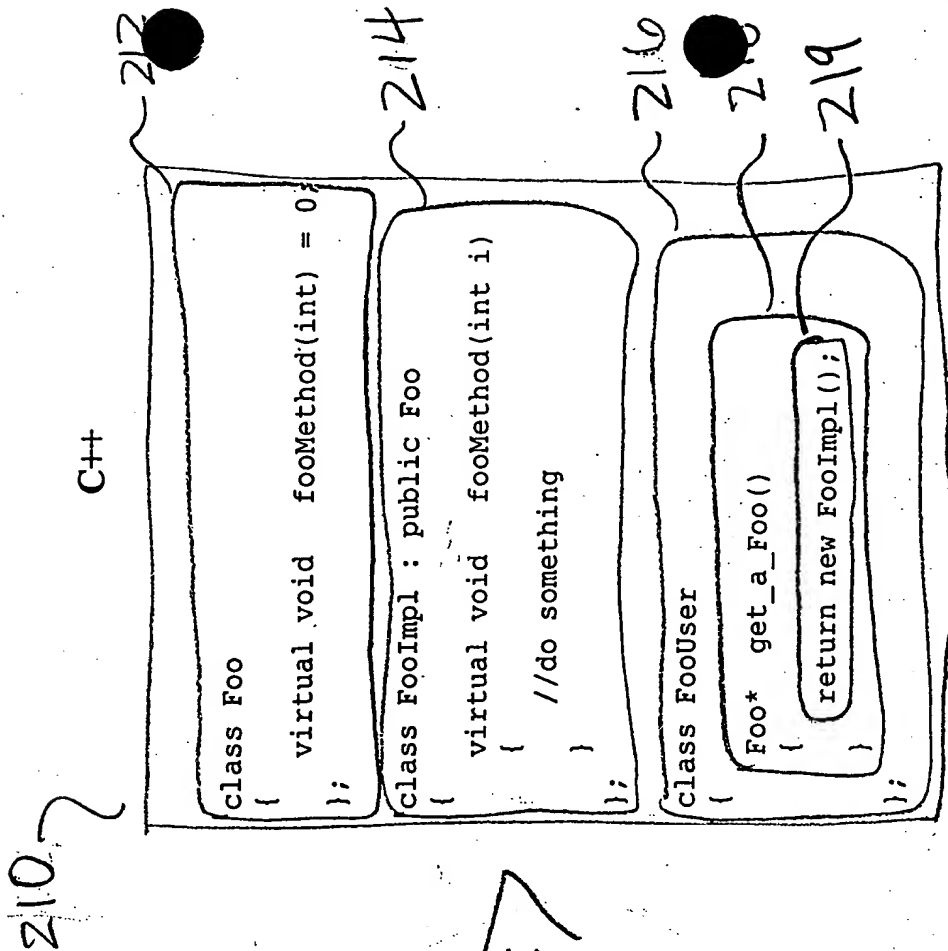


Fig. 4

C++

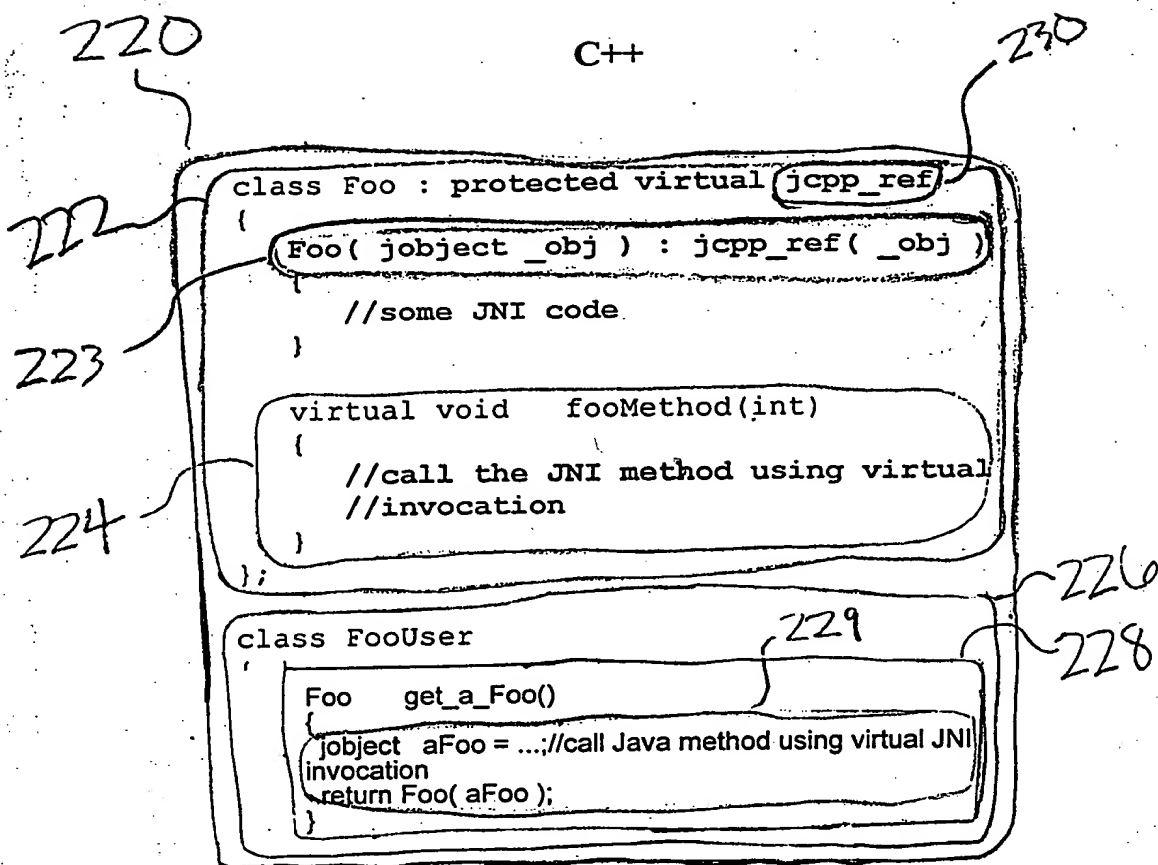


Fig. 5

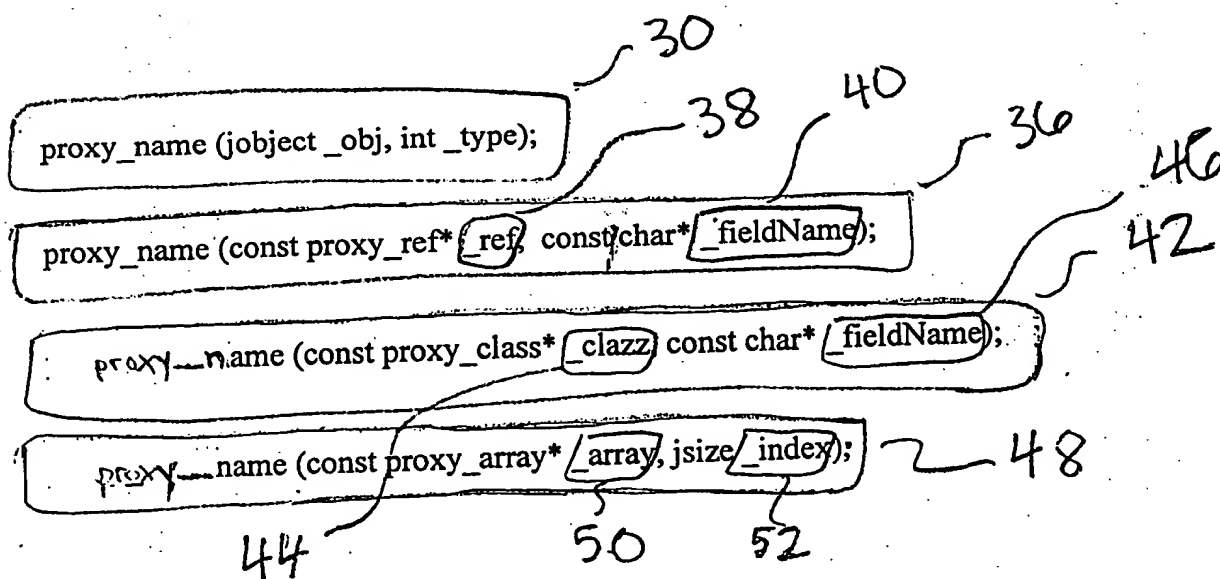


Fig. 6

002140" 94215560

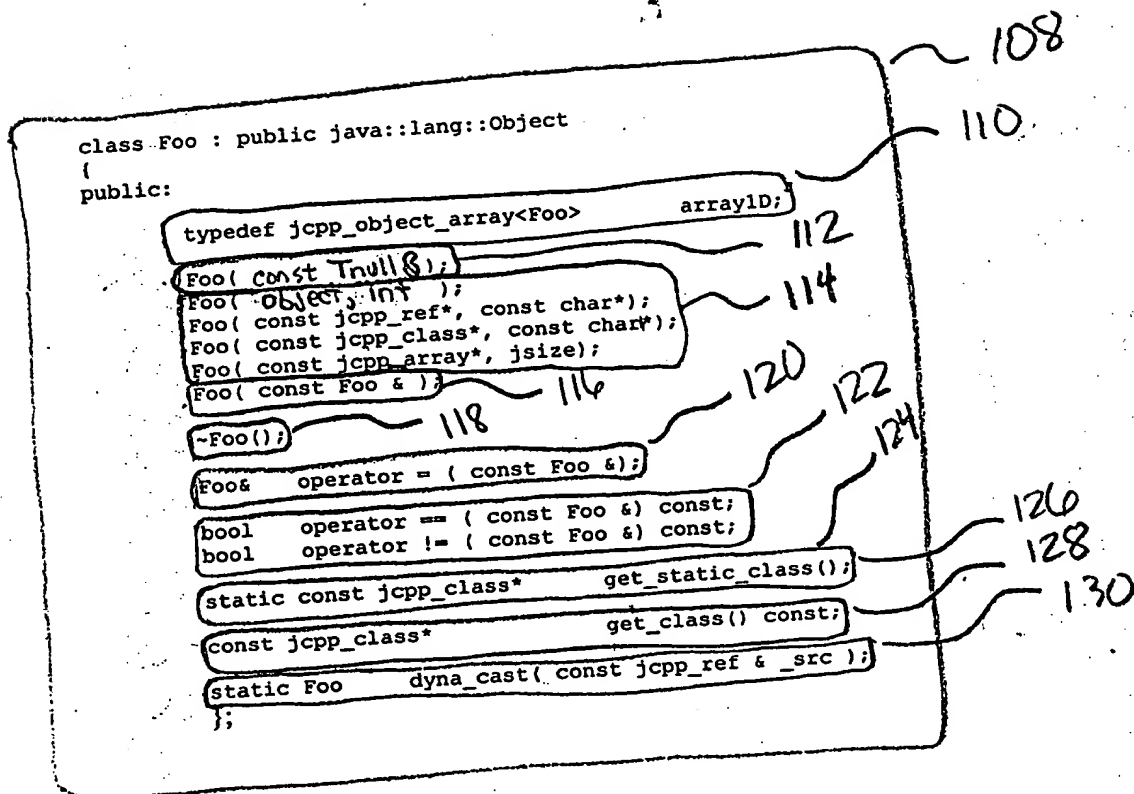


Fig. 7

52

```

class jcpp_int : public jcpp_base
{
public:
    typedef jcpp_int_array array1D;
    typedef Tobject_array<array1D> array2D;

    jcpp_int( const jcpp_ref * _ref, const char * _fieldName );
    jcpp_int( const jcpp_class * _ref, const char * _fieldName );
    jcpp_int( const jcpp_int_array * _array, jsize _index );
    jcpp_int( const jcpp_int & _rhs );
    jcpp_int( );

    operator new ( size_t _size );
    operator delete( void * _ptr );
    operator jint ( ) const;

    jcpp_int & operator = ( jint );
    jcpp_int & operator += ( jint );
    jcpp_int & operator -= ( jint );
    jcpp_int & operator *= ( jint );
    jcpp_int & operator /= ( jint );
    jcpp_int & operator %= ( jint );
    jcpp_int & operator ++ ( );
    jcpp_int & operator -- ( );
    jint operator ++ ( int );
    jint operator -- ( int );

    const jcpp_class * get_class() const;
};

```

Fig. 8

230

Java

Declarations

```
interface Foo {
    public static final int X=25;
}
```

Class Bar

```
{
    static int X=25;
    char ch = 'a';
}
```

Usage

```
int i = Foo.X; ~236
int j = Bar.X; ~238
Bar.X = 30; ~240
char k = new Bar().ch; ~242
Bar bar = new Bar(); ~244
bar.ch = 'd'; ~246
```

Fig. 9

C++

Declarations

```
class Foo {
    static const int X;
};
```

class Bar

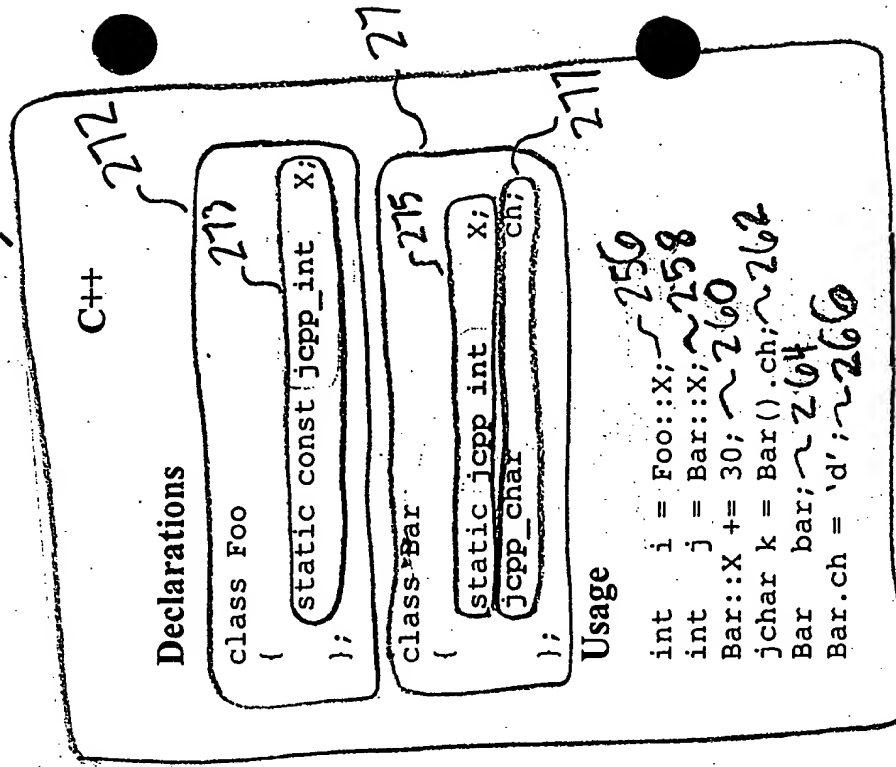
```
{
    static int X;
    char ch;
};
```

Usage

```
int i = Foo::X; ~256
int j = Bar::X; ~258
Bar::X = 30; ~260
char k = Bar().ch; ~262
Bar bar; ~264
Bar.ch = 'd'; ~266
```

Fig. 10

270



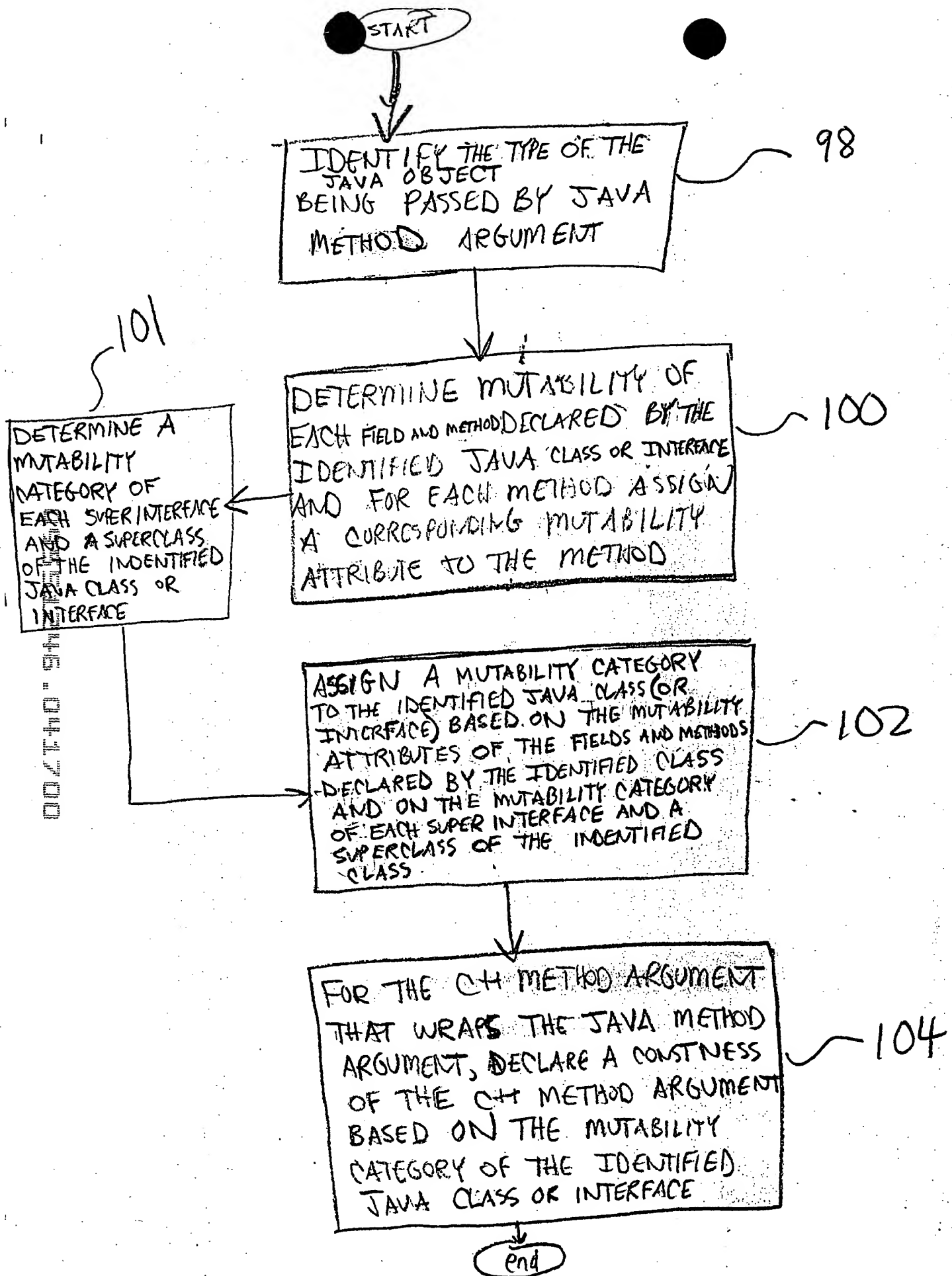


Fig 12.

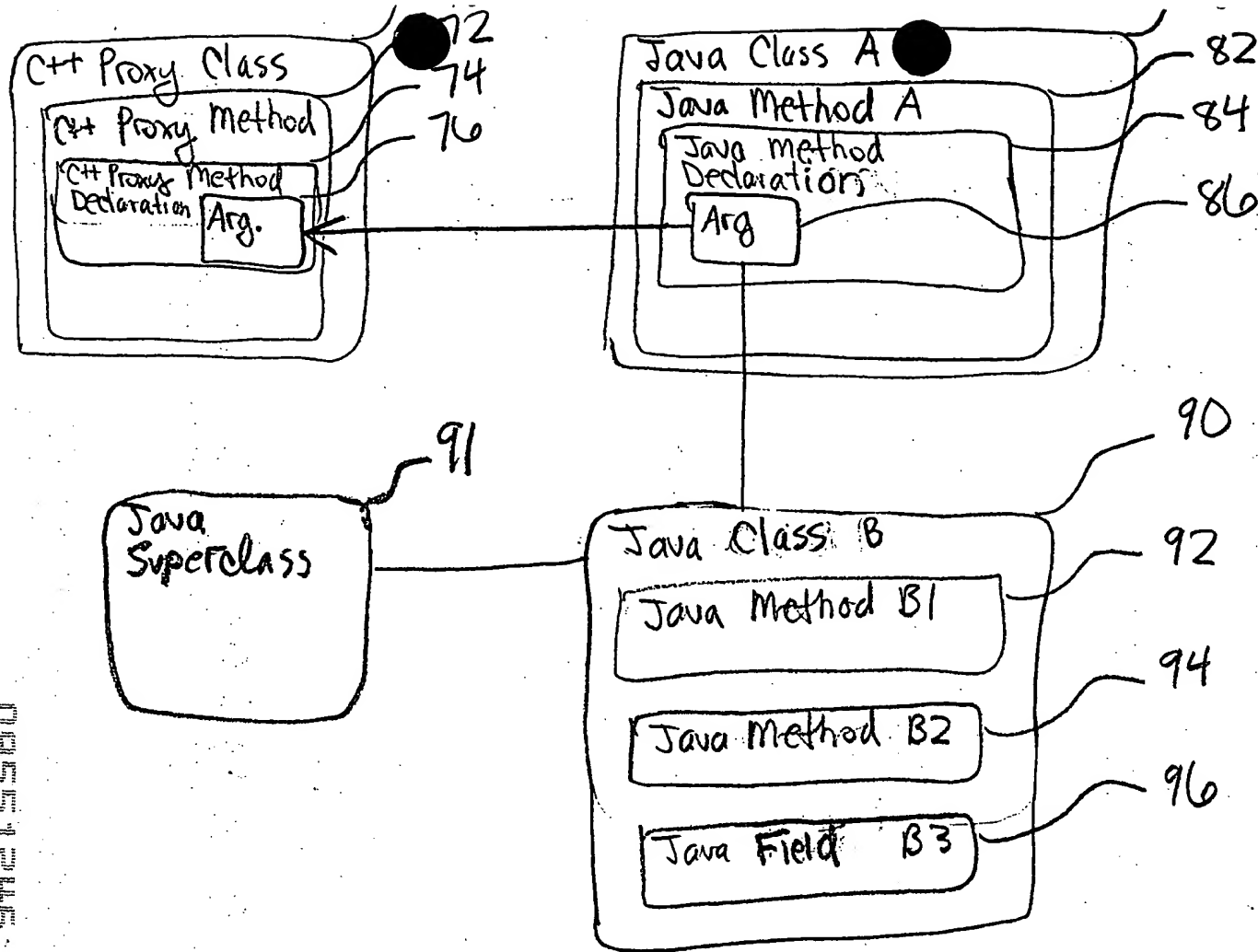


Fig 13

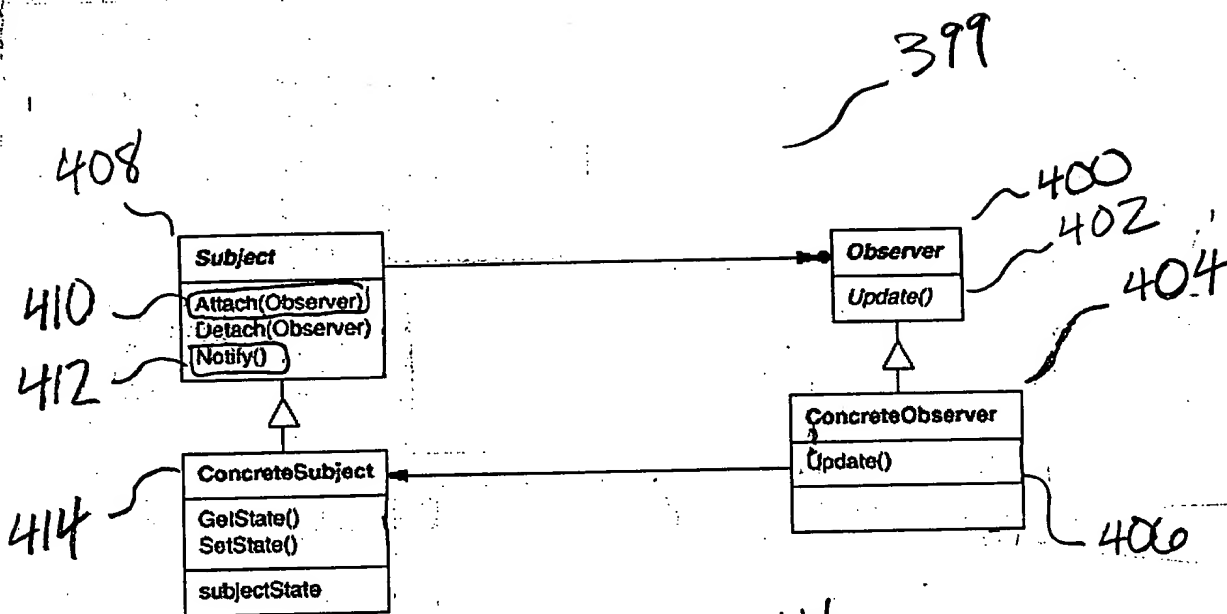


Fig 14

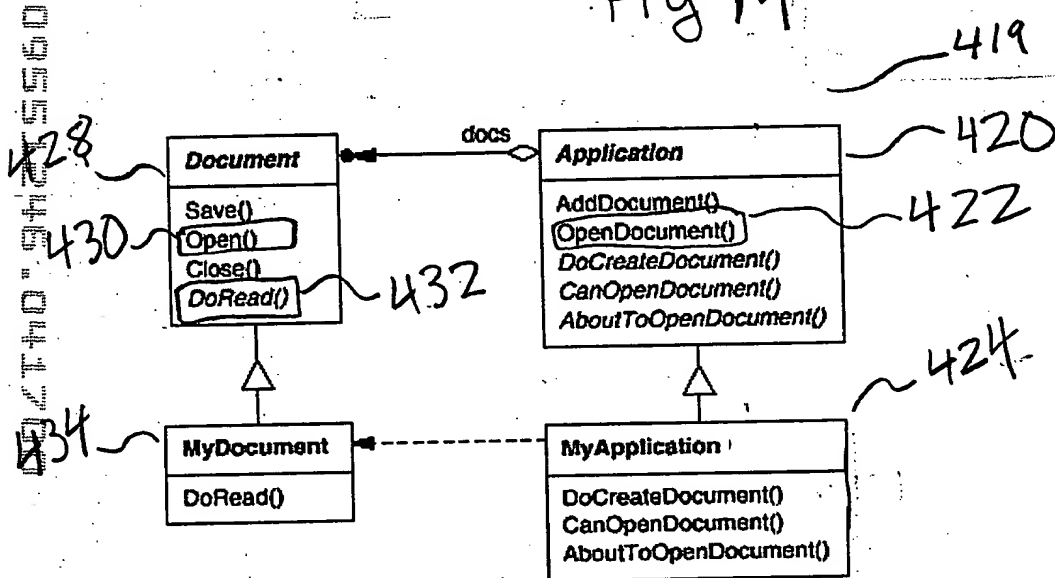


Fig 15

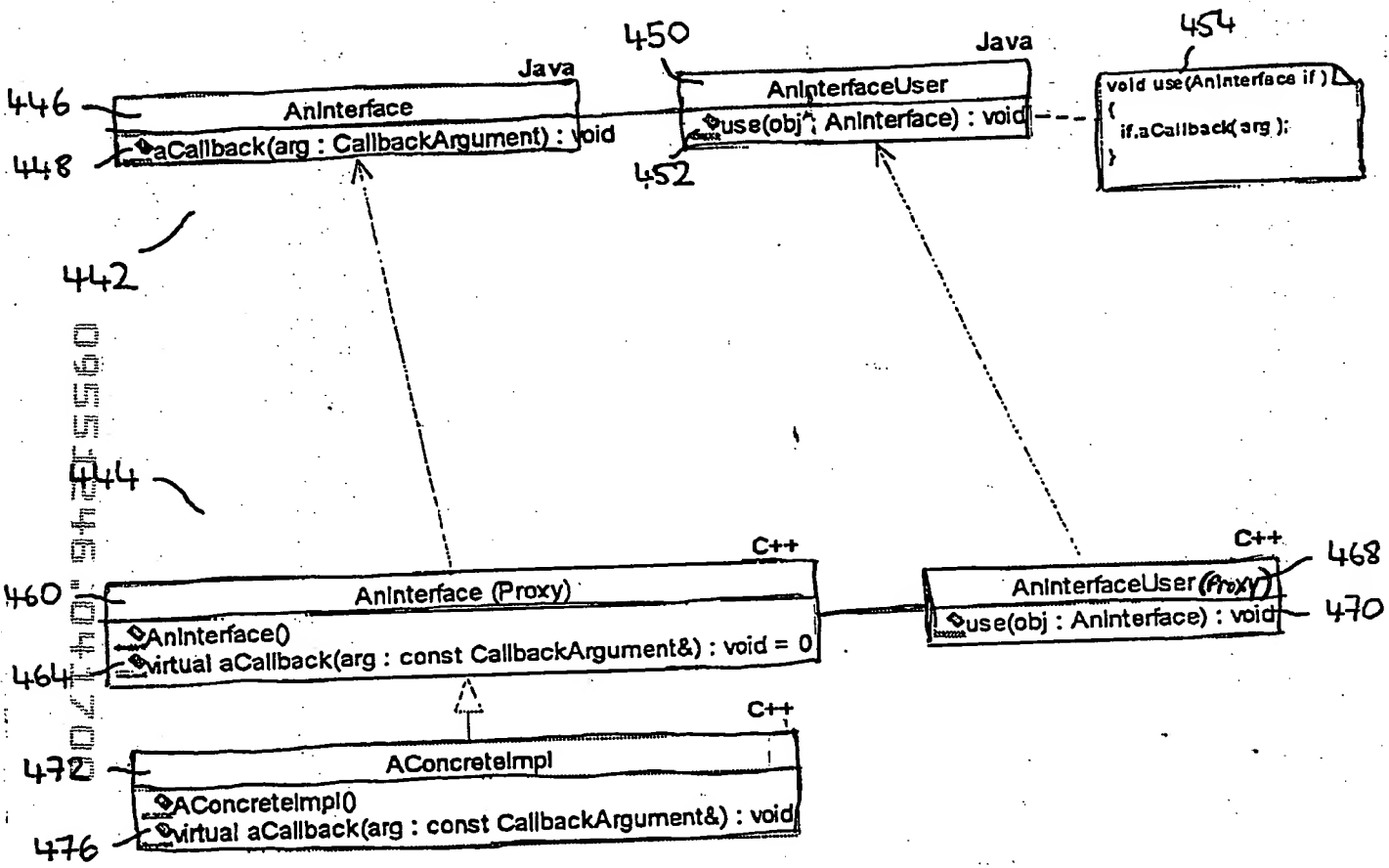


Fig 16

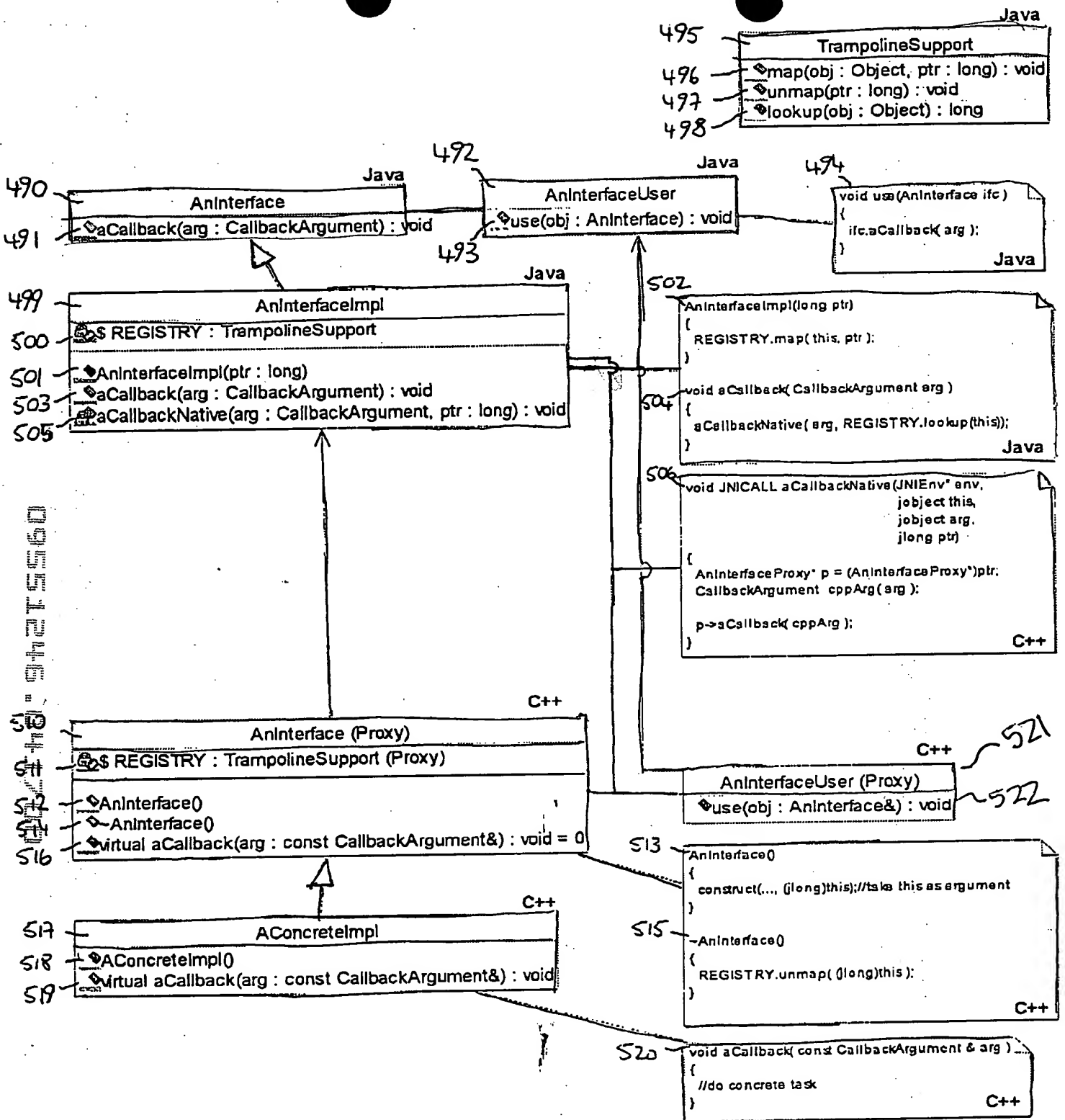


Fig. 17

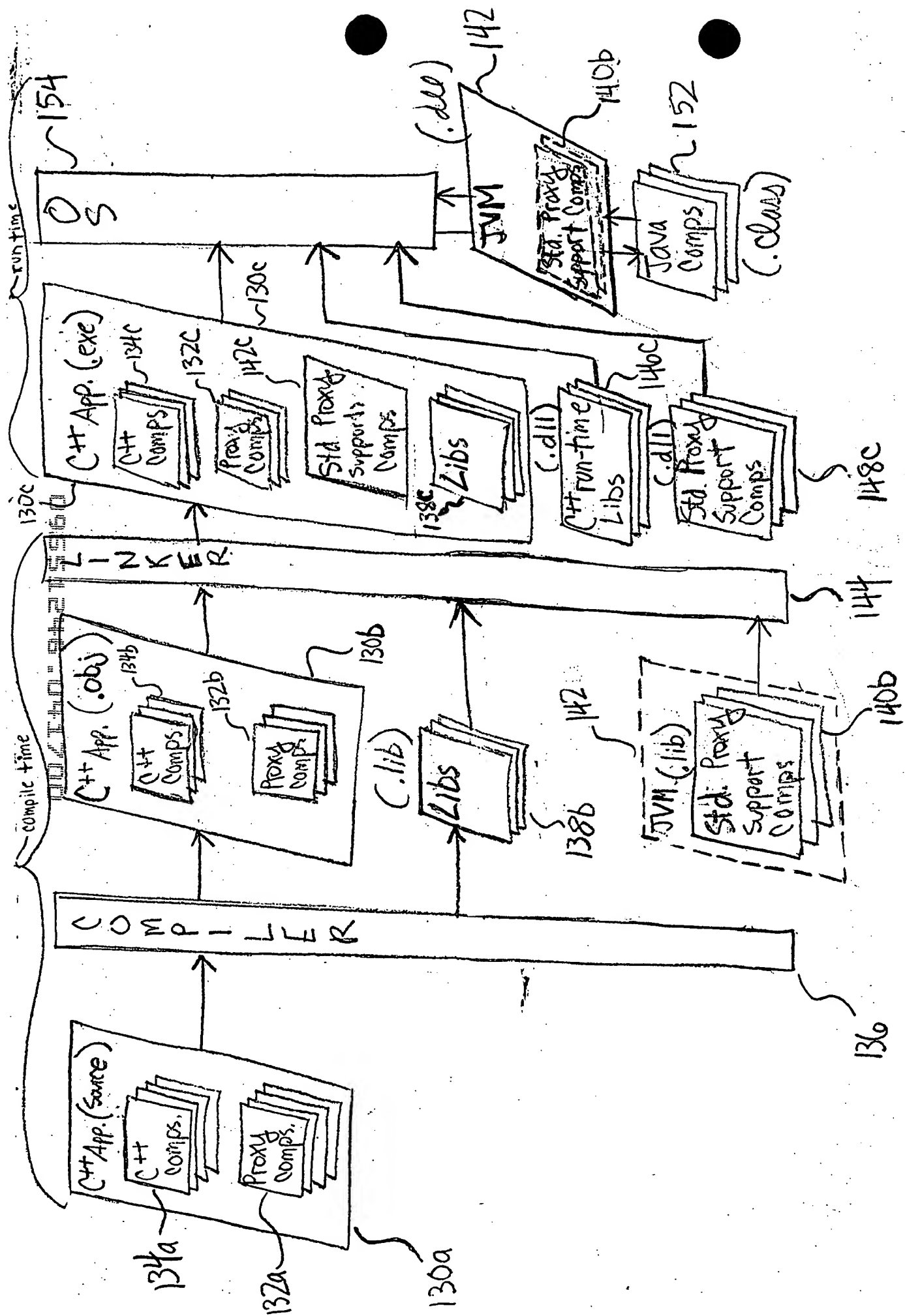


Fig. 18

540

542

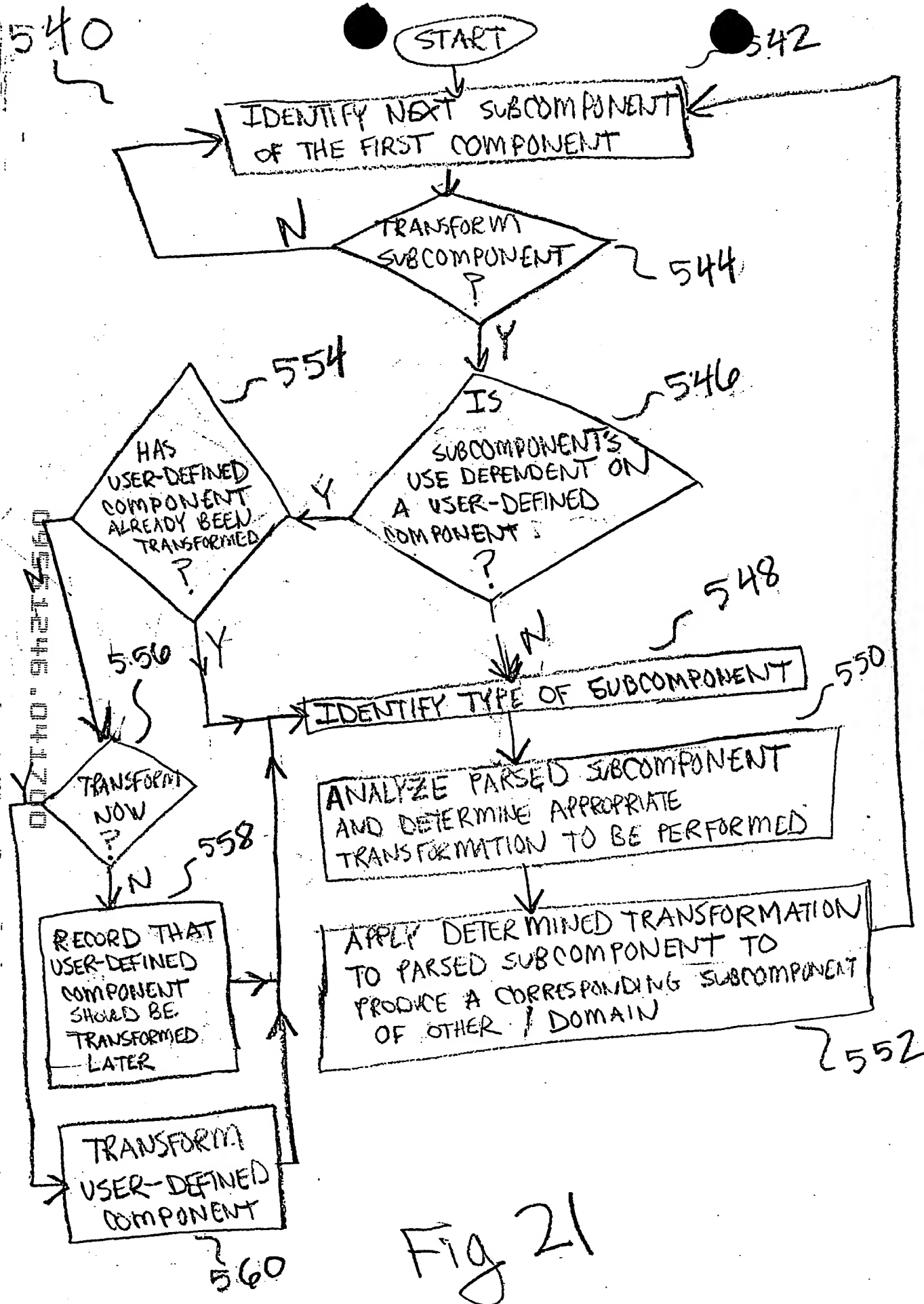


Fig 21

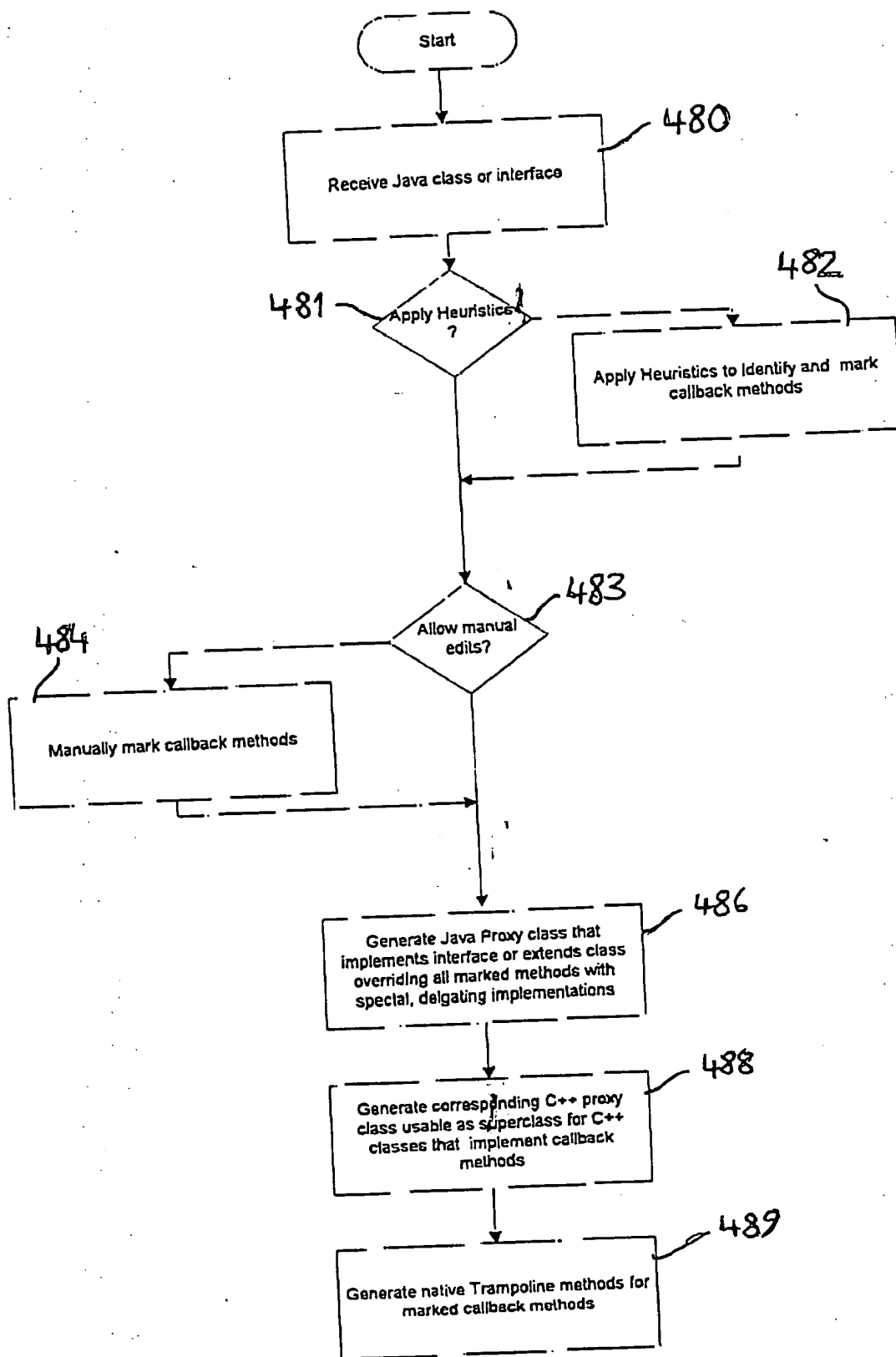


Fig. 22

300a

```
public class Counter implements java.io.Serializable
```

```
{
    public static final int    UP = 1;
    public static final int    DOWN = 2;
```

```
    private int    max;
    private int    direction;
```

```
    //creates a new UP-counter with the given maximum
    public Counter( int _max )
```

```
{
        this( _max, UP );
```

```
    //creates a new counter with given maximum and direction
    public Counter( int _max, int _direction )
```

```
{
        max = _max;
        direction = _direction;
```

```
    //counts in the direction specified and outputs the numbers
    public void    count()
```

```
{
        if( direction == UP )
            for( int I=0; I<max; I++ )
                System.out.println( " " + I );
        else if( direction == DOWN )
            for( int I=max-1; I>=0; I-- )
                System.out.println( " " + I );
```

```
    //returns true if this instance is an UP counter
    public boolean    isUpCounter()
```

```
{
        return ( direction == UP );
```

```
    //returns the maximum of the counter
    public final int    getMax()
```

```
{
        return max;
```

```
    //creates a counter with the same maximum as this counter, but reverse direction
    public Counter    getReverseCounter()
```

```
{
        return new Counter( max, direction == UP ? DOWN : UP );
```

Fig. 24

09551245-041700

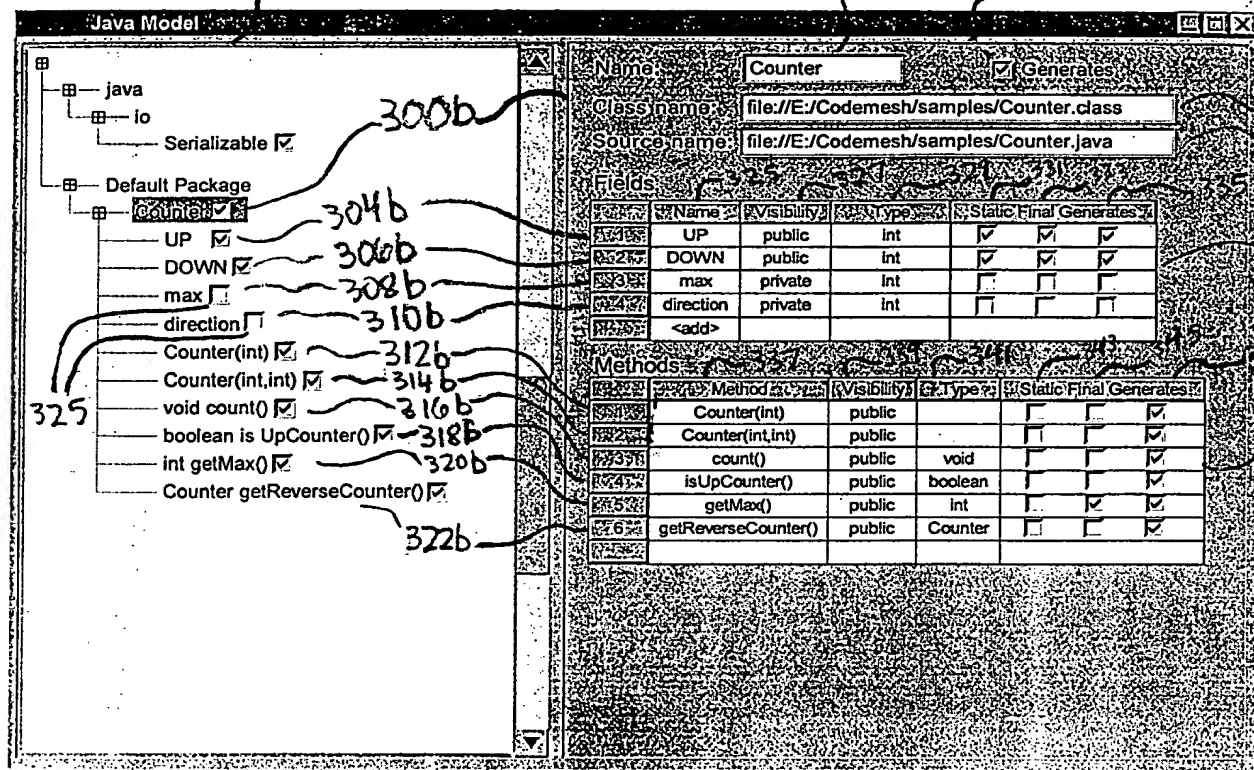


Fig. 25

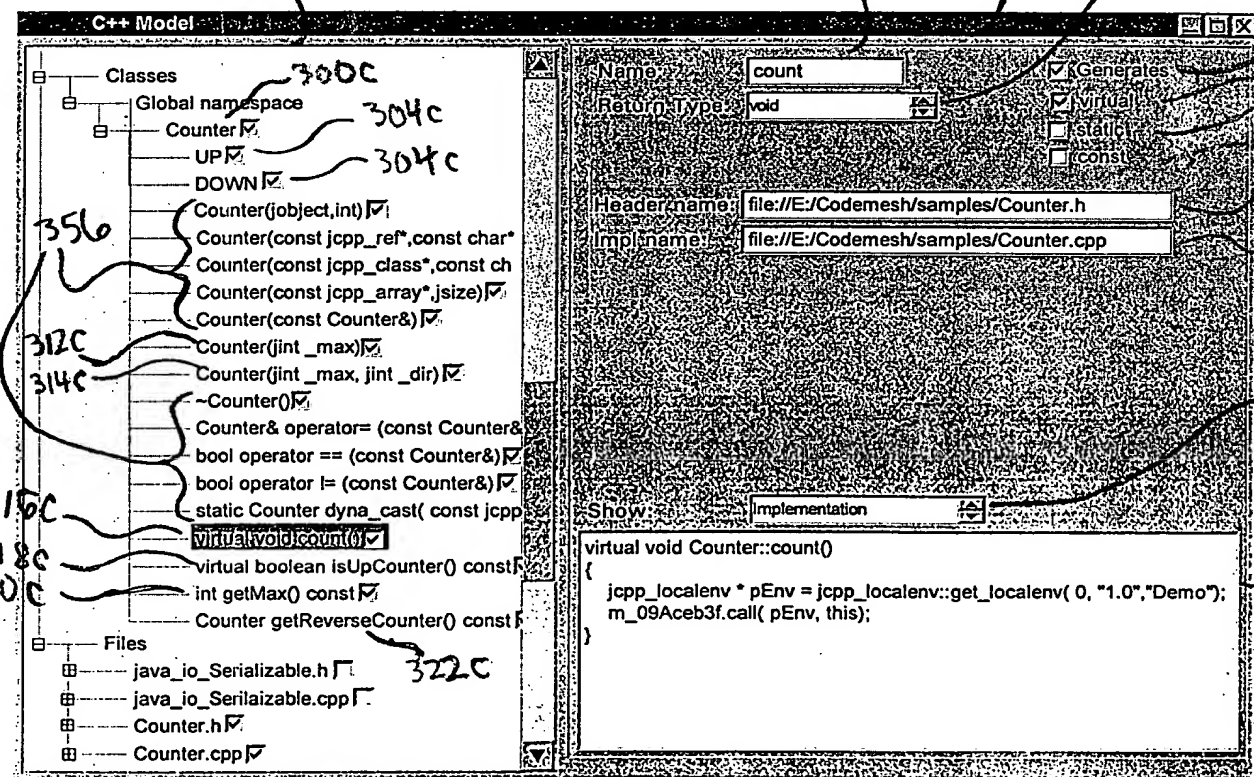


Fig. 26

Bottom-Up Port By Proxy (Initial)

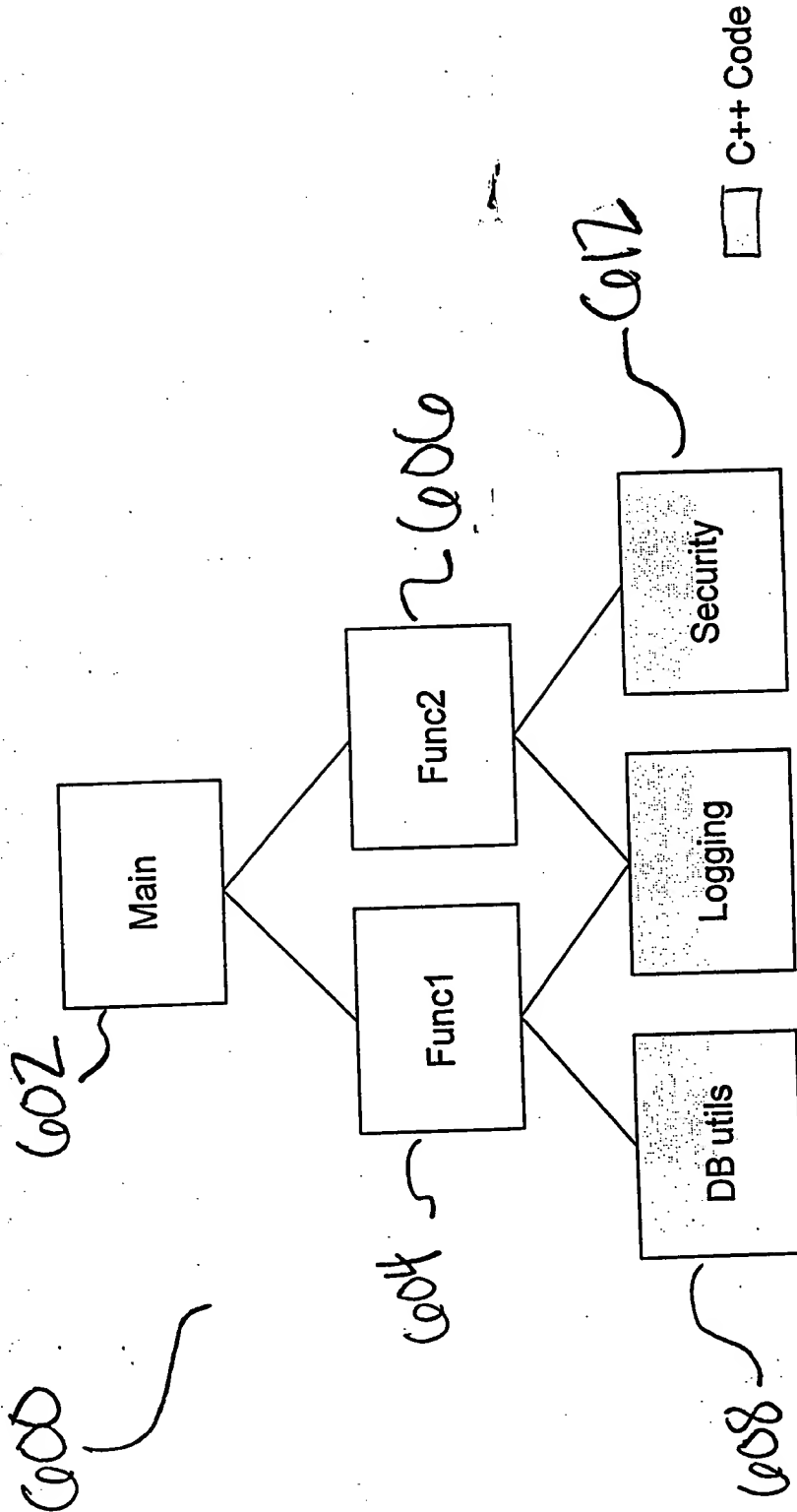
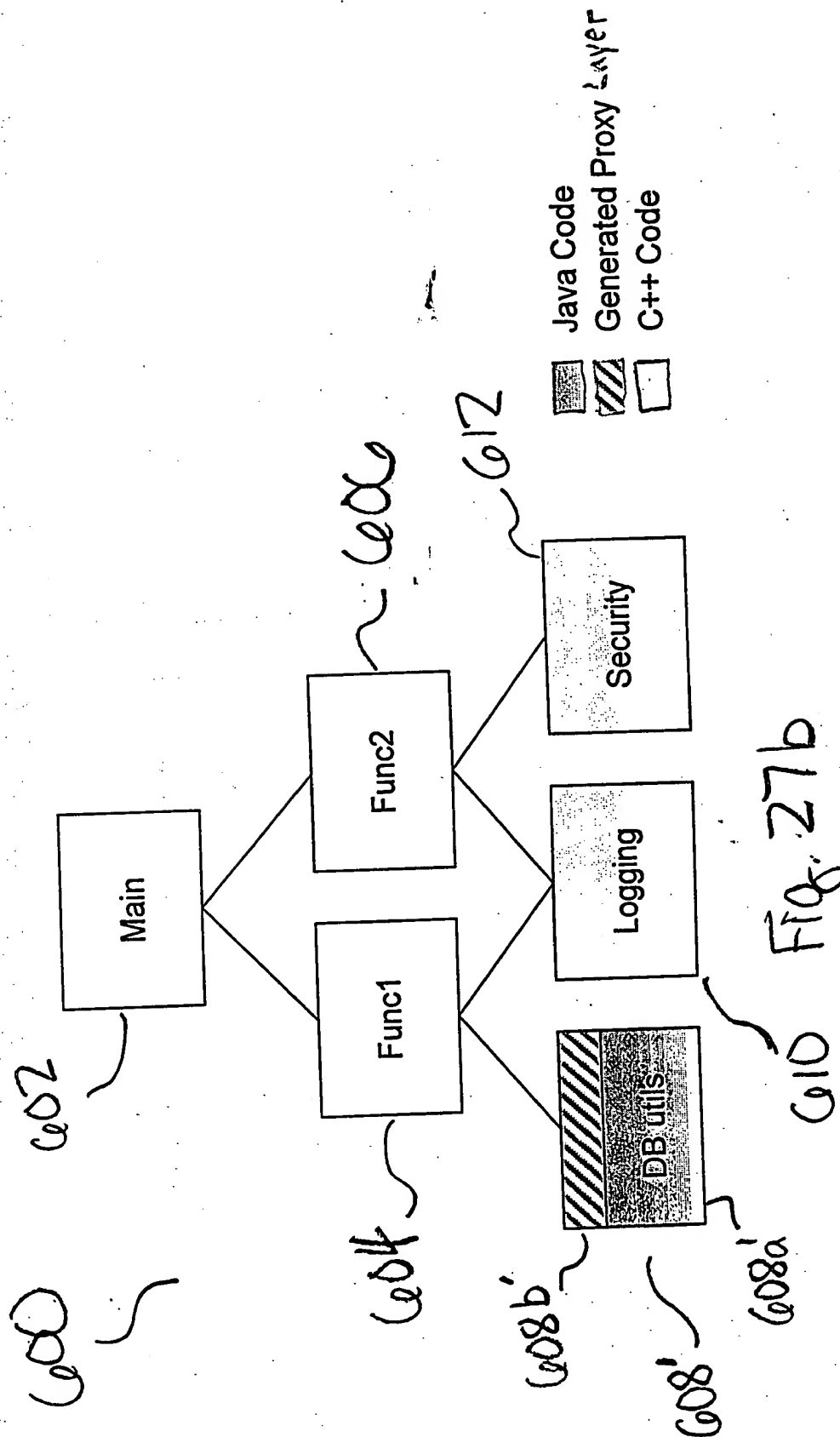


Fig. 27a

Bottom-Up Port By Proxy (1st Step)



Bottom-Up Port By Proxy (2nd Step)

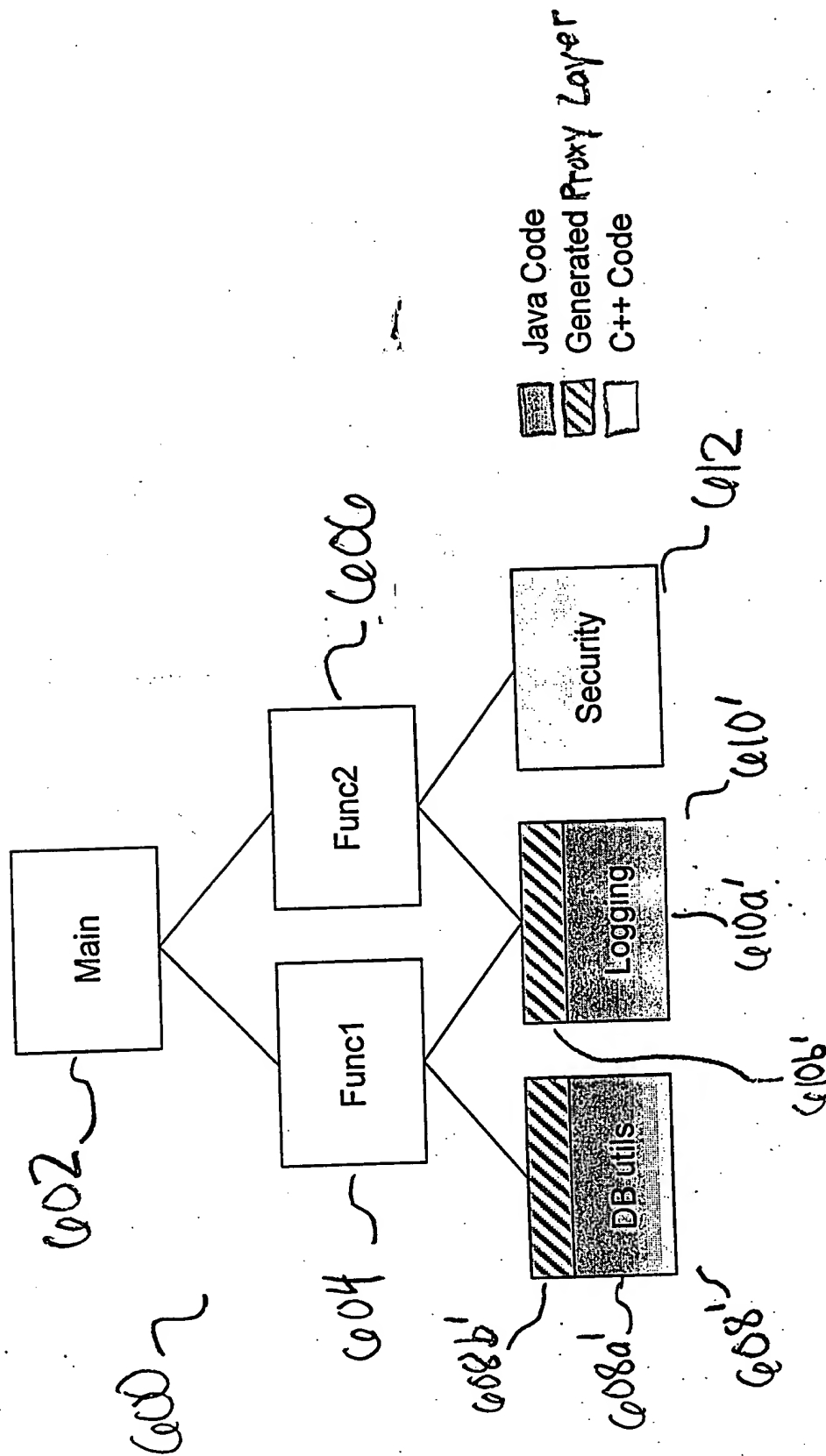


Fig. 27C

Bottom-Up Port By Proxy (3rd Step)

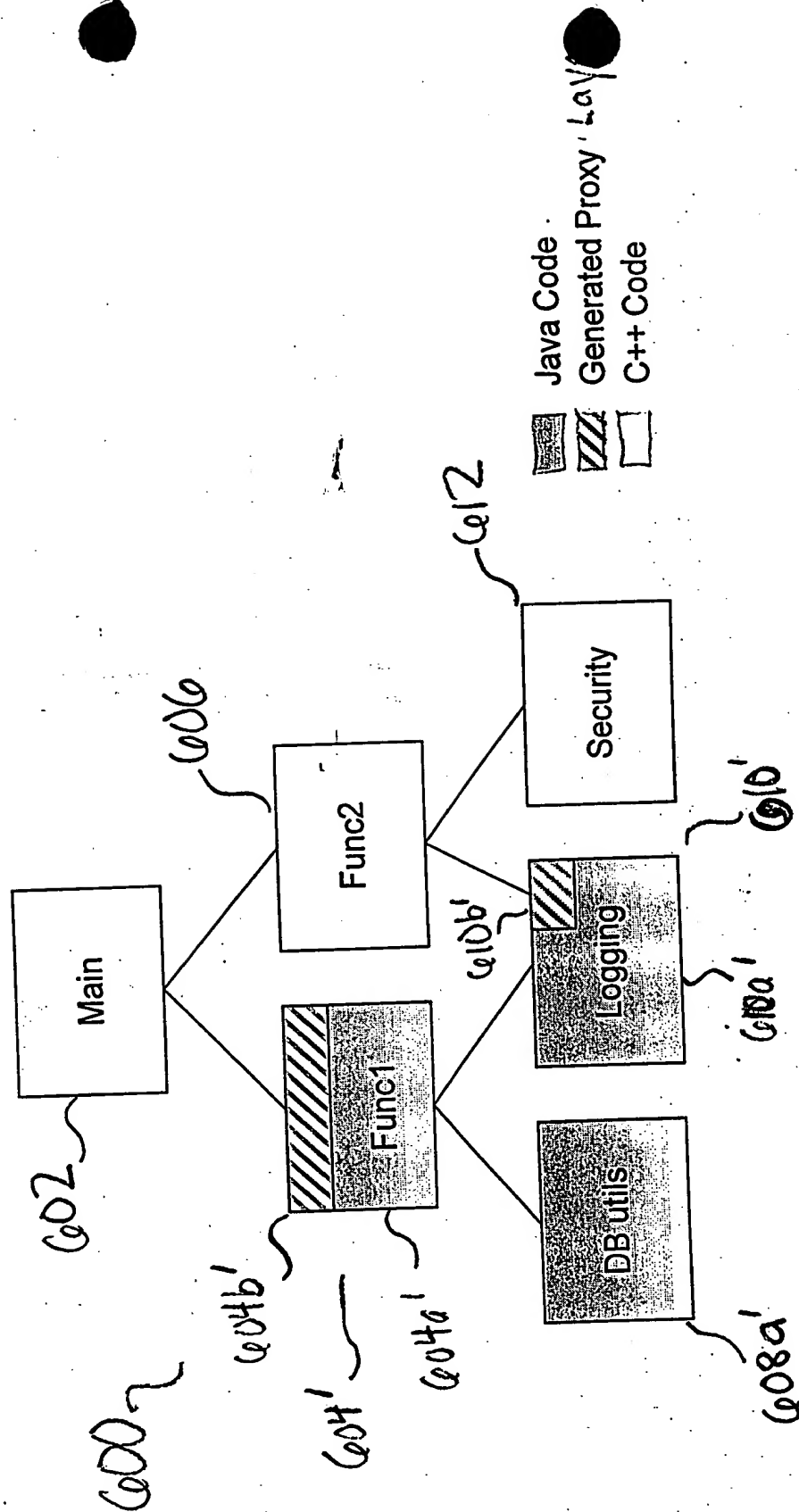


Fig. 27d

Bottom-Up Port By Proxy (4th Step)

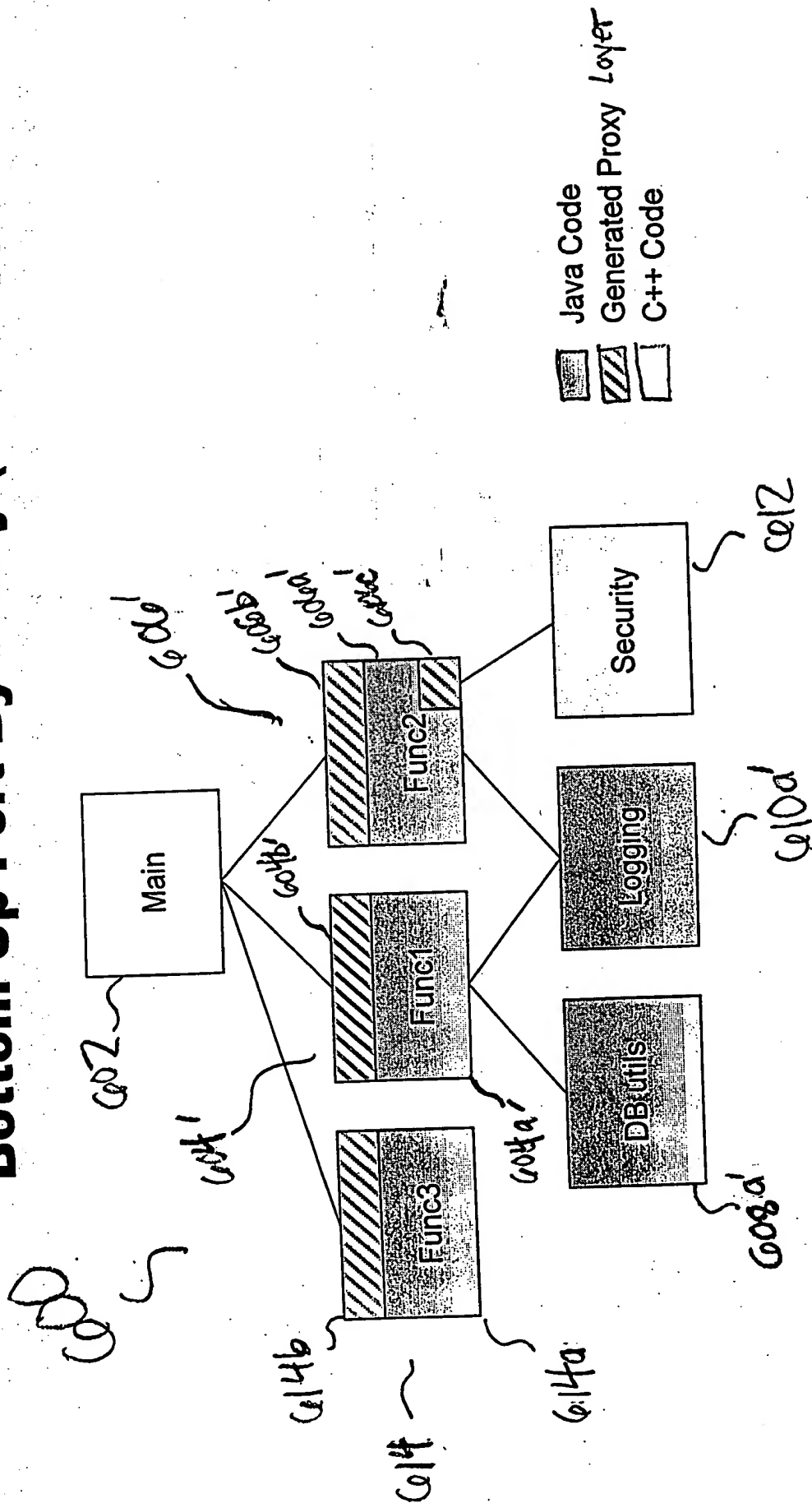


Fig. 27e

Bottom-Up Port By Proxy (5th Step)

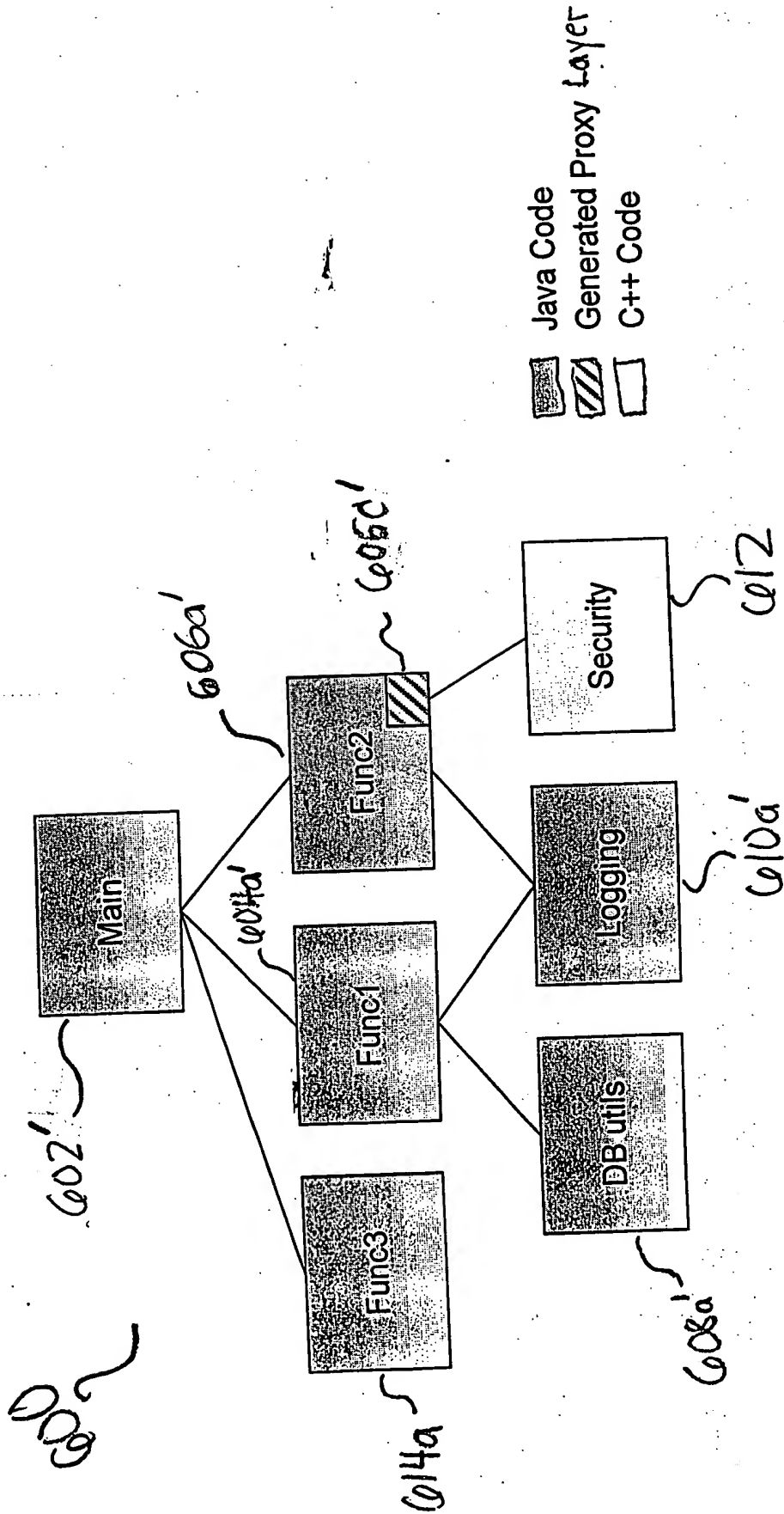


Fig. 27f